Multi-objective optimization using Deep Gaussian Processes: Application to Aerospace Vehicle Design

Ali Hebbal, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, Nouredine Melab

Multi-objective optimization using Deep Gaussian Processes: Application to Aerospace Vehicle Design

Ali Hebbal *
ONERA, DTIS, Université Paris Saclay, Université de Lille, CNRS/CRIStAL, Inria Lille

Loic Brevault† and Mathieu Balesdent ‡
ONERA, DTIS, Université Paris Saclay, F-91123 Palaiseau Cedex, France

El-Ghazali Talbi § and Nouredine Melab ¶
Université de Lille, CNRS/CRIStAL, Inria Lille, Villeneuve d'Ascq, France

This paper is focused on the problem of constrained multi-objective design optimization of aerospace vehicles. The design of such vehicles often involves disciplinary legacy models considered as black-box and computationally expensive simulations characterized by a possible non-stationary behavior (an abrupt change in the response or a different smoothness along the design space). The expensive cost of an exact function evaluation makes the use of classical evolutionary multi-objective algorithms not tractable. While Bayesian Optimization based on Gaussian Process regression can handle the expensive cost of the evaluations, the non-stationary behavior of the functions can make it inefficient. A recent approach consisting of coupling Bayesian Optimization with Deep Gaussian Processes showed promising results for single-objective non-stationary problems. This paper presents an extension of this approach to the multi-objective context. The efficiency of the proposed approach is assessed with respect to classical optimization methods on an analytical test-case and on an aerospace design problem.

I. Nomenclature

$(\mathcal{X}, \mathcal{Y}, C)$	=	Design of Experiment (DoE)
N	=	size of the DoE
n	=	number of objectives
n_c	=	number of constraints
D	=	Dimension of the input space
L	=	Number of layers in a Deep Gaussian Process
$\prec$	=	Pareto dominance relation
$\mathbf{x}^{(i)}$	=	i-th element of the DoE
x_i	=	i-th component of vector $\mathbf{x}$
BO	=	Bayesian Optimization
MO	=	Multi-Objective
$EHVI$	=	Expected HyperVolume Improvement
GP	=	Gaussian Process
DGP	=	Deep Gaussian Process
$\mathbf{h}_l$	=	l^{th} hidden unit
$\mathcal{Z}, \mathbf{u}$	=	Input-output induced variables
M	=	Number of induced inputs
$p(\cdot)$	=	Distribution of a variable
$q(\cdot)$	=	Approximated variational distribution

*Ph.D. student, ONERA, DTIS, Université Paris Saclay, Université de Lille, CNRS/CRIStAL, Inria Lille, ali.hebbal@onera.fr
†Research Engineer, ONERA, DTIS, Université Paris Saclay, loic.brevault@onera.fr
‡Research Engineer, ONERA, DTIS, Université Paris Saclay, mathieu.balesdent@onera.fr
§Professor at Polytech'Lille - University of Lille, el-ghazali.talbi@univ-lille1.fr
¶Professor at Polytech'Lille - University of Lille, nouredine.melab@univ-lille1.fr

II. Introduction

Aerospace vehicle design problems can ideally be modeled as multi-objective and multi-disciplinary optimization problems. In fact, different conflicting objectives need to be considered for aerospace vehicle design such as the payload mass, the gross lift-off weight, the availability or the production cost. In [3], a rich taxonomy of the applications of multi-objective optimization in aerospace engineering is presented. These multi-objective problems are characterized by n objectives that are optimized under n_c constraints in a D-dimensional design space (minimization is considered without loss of generality):

$$(P_{CMO}) \left| \begin{array}{lll} \text{Minimize}_{\mathbf{x}} & \mathbf{y} & = \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})] \\ \text{subject to} & g_i(\mathbf{x}) & \leq 0,\ i = 1, \ldots, n_c \end{array} \right. \tag{1}$$

where P_{CMO} stands for Constrained Multi-Objective problem
and $\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{X} \subseteq \mathbb{R}^D$
and $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Y} \subseteq \mathbb{R}^n$
$\mathbf{x}$ is called the decision vector, $\mathbb{X}$ the decision space, $\mathbf{y}$ the objective vector and $\mathbb{Y}$ the objective space.

One of the most used approaches to solve these problems are Multi-Objective Evolutionary Algorithms (MOEAs) [4]. Among the most popular MOEAs, NSGA-II (Non-dominated Sorting Genetic Algorithm II) [5] or SMPSO (Speed-constrained Multi-objective PSO)[6] can be cited. The advantage of these algorithms is that the use of a population-based search and diversity mechanisms makes it less prone to be trapped in local minima. Moreover, the use of simple operators for crossover and mutation allows the handling of highly non-linear or non-differentiable functions. However, MOEAs tend to need a consequent number of evaluations to converge to the exact Pareto front. This may make MOEAs not suitable for computationally expensive functions, where the concern is to minimize the number of evaluations. To overcome this issue, Bayesian Optimization (BO) is a widely used approach. It is based on surrogate models [7] that approximate the exact expensive functions allowing the evaluation of a greater number of design candidates. One of the most popular BO methods is "Efficient Global Optimization" (EGO) [8]. It uses the Gaussian Process (GP) regression [9] (also called Kriging) as surrogate models, providing an approximation of the objective and constraint functions and its associated uncertainty estimation. An acquisition function (or Infill Sampling Criterion) which uses these information given by the Gaussian Process regression models, is optimized to add the most promising point to the dataset. This point is then evaluated on the exact expensive functions and the surrogate models updated and so on, until a stopping criterion is satisfied. BO has been adapted to multi-objective optimization [10] by using new infill sampling criteria based on the concept of Pareto-Dominance as the Expected HyperVolume Improvement (EHVI) [11].

In many design optimization problems, the objective functions or the constraints are non-stationary. In fact, due to the abrupt change of some physical properties, the response may vary with a different smoothness along the input space. Specifically, in aerospace vehicle design optimization problems, the disciplines involved may present non-stationary behaviors. For example, in the structure discipline the stress–strain curve of a material can be non-stationary *i.e.* with a different trend in the elastic region, the strain hardening region and the necking region. In aerodynamics, computational fluid dynamics (CFD) problems, often have different specific flow regimes due to separation zones, circulating flows, vortex bursts, transitions from subsonic to transonic, supersonic and hypersonic conditions. GP regression is not adapted to predict these non-stationary functions since it is based on a stationary covariance function which implies a uniform smoothness of the prediction. To be able to approximate a non-stationary response using GP regression, different methods have been developed that can be classified into three categories:

- Direct formulation of non-stationary covariance function based on kernel convolution. The non stationary version of the squared exponential covariance function [12], and the Matérn covariance function [13] can be cited. The drawback of this approach is its difficulty to be applied to problem with dimension greater than 3 [13].
- The second approach consists in using local stationary covariance function. For example subdivising the input space into different subspaces where different stationary GPs are used [15] or the moving window approach where the training and prediction regions move along the input space [14]. However, the dataset size for a computationally expensive problem is limited and using a local surrogate model with sparser data may cause a poor approximation.
- Finally the non-linear mapping uses a parametrized function mapping between the input space and a new deformed space where the non-stationary function can be transformed into a stationary one. For example [16] propose a piece-wise density function with parametrized knots to map the input space with a deformed space. This method can show limitations when dealing with discontinued responses, or functions with non-stationarity not following linear directions.

Recently to handle the non-stationary issue in BO, the use of Deep Gaussian Processes (DGPs) has been proposed [1] [2] which is a class of surrogate models consisting of a functional composition of GPs [17]. DGPs show interesting results for handling non-stationary functions when coupled with BO for single objective problems [2].

The objective of this paper is to firstly generalize the coupling of BO with DGPs to the multi-objective case, and then apply the algorithm to a constrained multi-objective optimization of an aerospace vehicle design problem.
The paper is structured as follows. First, BO in the single and the multi-objective cases using GPs is briefly overviewed (Section III). Then, a description of DGP and its advantages over GP with a focus on its coupling with MO-BO is presented (Section IV). Next, experimentation on an analytical problem is performed to confirm the interest of the proposed approach (Section V). Finally, the paper concludes with the application to a multi-objective optimization of an aerospace vehicle design problem (Section V).

III. Bayesian-Optimization using Gaussian Processes

In this section a review on Bayesian Optimization using Gaussian Processes for the single and multi-objective case is presented.

A. Single-Objective Bayesian Optimization

A Bayesian Optimization Algorithm consists in a loop between a modeling procedure usually using a GP regression model and a sampling procedure using an infill sampling criterion (Fig. 1). A GP is completely defined by its mean function $\mu(\cdot)$ and covariance function $k(\cdot,\cdot)$. The covariance function is usually considered parametrized by a set of hyper-parameters Θ. The particularity of GP regression models is that for an uneavaluated candidate $\mathbf{x}^*$ along the prediction $\hat{y}^*$, a Gaussian error $\hat{\sigma}^*$ of this prediction is obtained, hence giving uncertainty information. The prediction with its uncertainty as Gaussian error are used to estimate the possible improvement offered by a new candidate with respect to the current optimum. This measure of improvement is called an Infill Sampling criterion or acquisition function which is optimized on the design space to determine the most promising candidate to add to the sample. One popular infill sampling criterion is the Expected Improvement (EI) [8] which computes the mathematical expected value of the improvement of a candidate. EI has been adapted to the constrained case via the use of the Probability of Improvement or the Expected Violation [18].

B. Multi-objective Bayesian Optimization

Bayesian algorithms have been extended to solve multi-objective optimization [19]. A variety of approaches have been proposed for MO-BO which can be classified into the aggregation-based method (using BO on a weighted sum of objective functions) [20] [21] and the dominance-based approach (using new infill sampling criteria based on the concept of Pareto-Dominance)[19] [22]. In this study, the second approach is used. It follows the same structure as Single-Objective Bayesian Optimization, with the difference that for each objective and constraint function, a surrogate model is created and it uses an infill sampling criterion based on the concept of Pareto-Dominance such as the Expected HyperVolume Improvement (EHVI) [11].

1. Definition of the Expected HyperVolume Improvement

The notion of Expected HyperVolume Improvement (EHVI) was first introduced by Emmerich *et al.* [19]. Let consider an unconstrained multi-objective problem and let $\mathbb{B}$ be a finite hypervolume of the objective space where all possible solutions lie. $\mathbb{B} = \left\{\mathbf{y} \in \mathbb{R}^n; \mathbf{y}^L \leq \mathbf{y} \leq \mathbf{y}^U\right\}$ where $\mathbf{y}^L$ and $\mathbf{y}^U$ are the chosen lower and upper bounds respectively. The exact objective functions $f_1(\cdot), \ldots, f_n(\cdot)$ are evaluated over a training sample set $\mathcal{X}_N = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right\}$ resulting in the evaluated sample $\mathcal{Y}_N = \left\{\mathbf{y}^{(1)} = \mathbf{f}\left(\mathbf{x}^{(1)}\right), \ldots, \mathbf{y}^{(N)} = \mathbf{f}\left(\mathbf{x}^{(N)}\right)\right\}$. The dominated hypervolume of the samples is defined as follows:

$$H_{\mathcal{Y}_N} = \left\{\mathbf{y} \in \mathbb{B}; \exists i \in \{1, \ldots, N\}, \mathbf{y}^{(i)} \prec \mathbf{y}\right\} \tag{2}$$

So $H_{\mathcal{Y}_N}$ is the subset of $\mathbb{B}$ whose points are dominated by the sample set. Let $\mathbf{x}^{(N+1)}$ be a new point added to the sample and $\mathbf{y}^{(N+1)}$ its evaluation on the exact objective functions. Since $H_{\mathcal{Y}_N} \subset H_{\mathcal{Y}_{N+1}}$, the hypervolume improvement of the sample set by adding $\mathbf{x}^{(N+1)}$ is given by: $I_{\mathcal{Y}_N}(\mathbf{x}_{N+1}) = |H_{\mathcal{Y}_N}| - |H_{\mathcal{Y}_{N+1}}|$ where $|\cdot|$ is the standard Lebesgue measure. Fig. 2 illustrates the concepts introduced previously in the two-objective case.

Let $Y_1(\cdot) \sim \mathcal{N}\left(\hat{y}_1(\cdot), \hat{\sigma}_1(\cdot)\right), \ldots, Y_n(\cdot) \sim \mathcal{N}\left(\hat{y}_n(\cdot), \hat{\sigma}_n(\cdot)\right)$ be the Gaussian process models of $f_1(\cdot), \ldots, f_n(\cdot)$ and $\mathbf{Y}(\cdot)$ the vector $\mathbf{Y}(\cdot) = (Y_1(\cdot), \ldots, Y_n(\cdot))$. The Expected HyperVolume Improvement for a point $\mathbf{x}$ is then defined as the

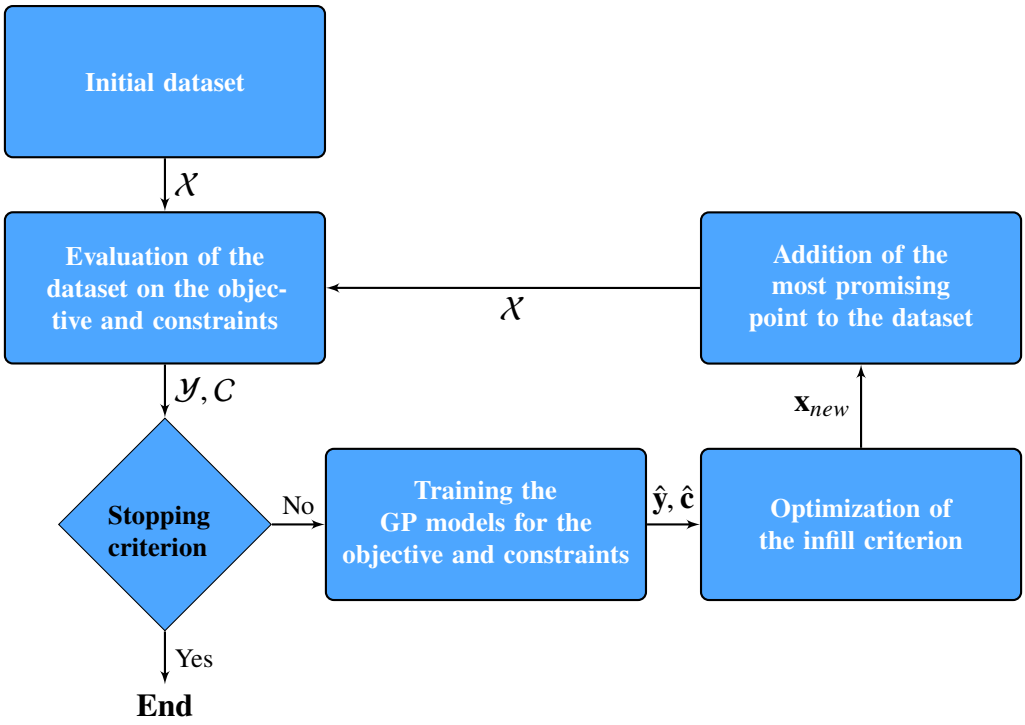


Fig. 1 Bayesian Optimization framework for single-objective problems

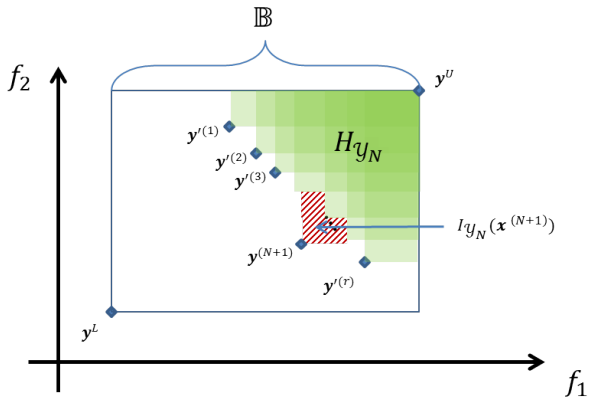


Fig. 2 Example of an improvement of the dominated hypervolume in the two-objective case

mathematical expected value of hypervolume improvement by adding this point to the sample set, it is derived as:

$$EHVI_{\mathcal{Y}_N}(\mathbf{x}) = \mathbb{E}\left(|H_{\mathcal{Y}_{N+1}}| - |H_{\mathcal{Y}_N}|\right) = \int_{\mathbb{B}\setminus H_{\mathcal{Y}_N}} p\left(\mathbf{Y}(\mathbf{x}) \prec \mathbf{u}\right) \mathrm{d}\mathbf{u} \tag{3}$$

2. Computation of the EHVI in the two-objective case

The computation of the EHVI for many objectives is a non-trivial problem. Several methods [23] [24] have been proposed to compute the EHVI formula, however, the computational complexity increases exponentially with the number of objectives. In this study the number of objectives is restrained to two.

First the objective functions are assumed to be independent so $p(\mathbf{Y}(\mathbf{x}) \prec \mathbf{u}) = p(Y_1(\mathbf{x}) \prec u_1) \times p(Y_2(\mathbf{x}) \prec u_2)$

Let $\left(\mathbf{x}'^{(1)}, ..., \mathbf{x}'^{(r)}\right)$ be the set of the non dominated points of the sample set and $\left(\mathbf{y}'^{(1)}, \ldots, \mathbf{y}'^{(r)}\right)$ the corresponding

function values $\mathbf{y}'^{(i)} = \mathbf{f}(\mathbf{x}'^{(i)}) = \left[f_1\left(\mathbf{x}'^{(i)}\right), f_2\left(\mathbf{x}'^{(i)}\right)\right]$ sorted in ascending order of the value of the objective function $f_1(\cdot)$. In the objective space, the hypervolume $\mathbb{B} \setminus H_{\mathcal{Y}_N}$ is splited into $r+1$ rectangles R_t with $t \in \{1, \ldots, r+1\}$ (Fig. 3). Each rectangle R_t is delimited horizontally by $y'^{(t-1)}_1$ and $y'^{(t)}_1$ and vertically by $y'^{(t-1)}_2$ and $y'^{(0)}_2$. With $\mathbf{y}'^{(0)} = \mathbf{y}^L$ and $\mathbf{y}'^{(r+1)} = \mathbf{y}^U$. Hence the integration domain $\mathbb{B} \setminus H_{\mathcal{Y}_N}$ is partitioned into rectangles completely defined, over which the integral can be decomposed. Therefore, Eq. 3 can be rewritten:

$$
\begin{aligned}
EHVI_{\mathcal{Y}_N}(\mathbf{x}) &= \int_{\mathbf{u}\in\mathbb{B}\setminus H_{\mathcal{Y}_N}} p(\mathbf{Y}(\mathbf{x}) \prec \mathbf{u})\mathrm{d}\mathbf{u} \\
&= \int\int_{\mathbf{u}=(u_1,u_2)\in\mathbb{B}\setminus H_{\mathcal{Y}_N}} p(Y_1(\mathbf{x}) \prec u_1)p(Y_2(\mathbf{x}) \prec u_2)\mathrm{d}u_1\mathrm{d}u_2 \\
&= \sum_{t=1}^{r+1} \int_{y'^{(t-1)}_1}^{y'^{(t)}_1} p(Y_1(\mathbf{x}) \prec u_1) \int_{y'^{(0)}_2}^{y'^{(t-1)}_2} p(Y_2(\mathbf{x}) \prec u_2)\mathrm{d}u_1\mathrm{d}u_2 \\
&= \sum_{t=1}^{r+1} \int_{y'^{(t-1)}_1}^{y'^{(t)}_1} \Phi\left(\frac{u_1 - \hat{y}_1(\mathbf{x})}{\hat{\sigma}_1(\mathbf{x})}\right) \int_{y'^{(0)}_2}^{y'^{(t-1)}_2} \Phi\left(\frac{u_2 - \hat{y}_2(\mathbf{x})}{\hat{\sigma}_2(\mathbf{x})}\right)\mathrm{d}u_1\mathrm{d}u_2
\end{aligned} \tag{4}
$$

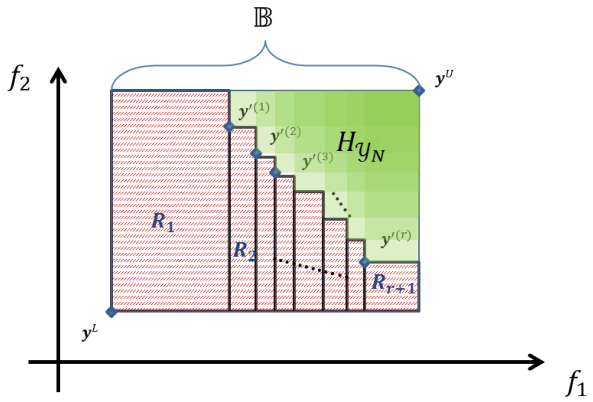


Fig. 3 Illustration of the decomposition of the objective space

The integration of the Cumulative Distribution Function (CDF) of a Gaussian distribution $\Phi(\cdot)$ comes back to the integration of the error function which is a tractable analytic computation. The computation of the EHVI in the two-objective case can be implemented analytically.

In the constrained case, the same adaptation than in the single objective BO may be adopted. Specificaly, by considering the EHVI of the feasible solutions and a constraint infill criterion as the probability of feasibility or the expected violation [18] to combine with the EHVI.

IV. Bayesian Optimization using Deep Gaussian Processes

In this section Deep Gaussian Processes are described, with a discussion on their coupling with BO for the single and multi-objective cases.

A. Deep Gaussian Processes

A DGP [17] is a deep network architecture where each layer is a GP. In fact, a DGP is a nested structure of GPs considering the relationship between the inputs and the final output as a functional composition of GPs (Fig. 4):

$$
y = f_L(\mathbf{f}_{L-1}(\ldots\mathbf{f}_l(\ldots(\mathbf{f}_1(\mathbf{f}_0(\mathbf{x}) + \epsilon_0) + \epsilon_1)\ldots) + \epsilon_l)\ldots + \epsilon_{L-1}) + \epsilon_L \tag{5}
$$

where L is the number of layers, $\mathbf{f}_l(\cdot)$ is an intermediate GP and $\epsilon_l \sim \mathcal{N}(0, \sigma_l^2\mathbf{I})$ is a Gaussian noise introduced in each layer. Each layer l is composed of an input node $\mathbf{h}_l$, an output node $\mathbf{h}_{l+1}$ and a GP $\mathbf{f}_l(\cdot)$ mapping between the two nodes, getting the recursive equation: $\mathbf{h}_{l+1} = \mathbf{f}_l(\mathbf{h}_l) + \epsilon_l$. $\mathbf{h}_l$, $\mathbf{h}_{l+1}$ and $\mathbf{f}_l(\cdot)$ can be multidimensional, in this case for each component $h_{l+1,i}$ of $\mathbf{h}_{l+1}$ a GP $f_{li}(\cdot)$ maps between $\mathbf{h}_l$ and $h_{l+1,i}$ (Fig. 5).

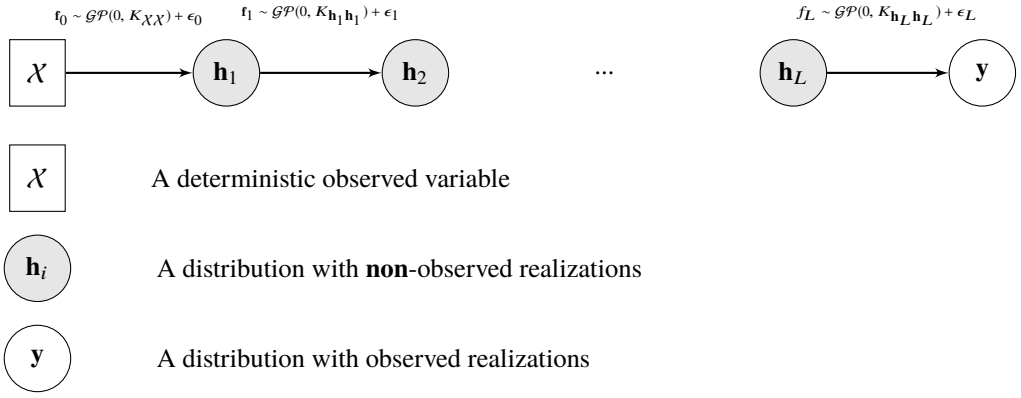


Fig. 4 A representation of the structure of a DGP

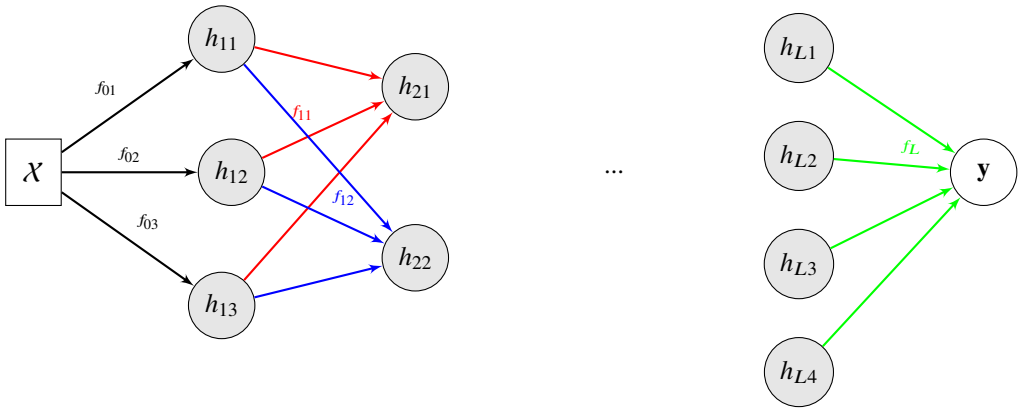


Fig. 5 Example of an exploded view of the structure of a DGP

As in GP regression, for training the DGP model, the marginal likelihood $p(\mathbf{y}|\mathcal{X})$ is maximized using an optimization algorithm (Eq. 7).

$$p(\mathbf{y}|\mathcal{X}) = \int_{\mathbf{h}_1} \dots \int_{\mathbf{h}_l} \dots \int_{\mathbf{h}_L} p(\mathbf{y}, \mathbf{h}_1, \dots, \mathbf{h}_l, \dots, \mathbf{h}_l|\mathcal{X})\, \mathrm{d}\mathbf{h}_1 \dots \mathrm{d}\mathbf{h}_L = \int_{\{\mathbf{h}_l\}_1^L} p\left(\mathbf{y}, \{\mathbf{h}_l\}_1^L|\mathcal{X}\right) \mathrm{d}\{\mathbf{h}_l\}_1^L \tag{6}$$
$$= \int_{\{\mathbf{h}_l\}_1^L} p(\mathbf{y}|\mathbf{h}_L)p(\mathbf{h}_L|\mathbf{h}_{L-1})...p(\mathbf{h}_1|\mathcal{X})\mathrm{d}\{\mathbf{h}_l\}_1^L$$

where $\{\mathbf{h}_l\}_1^L$ is the set of hidden layers $\{\mathbf{h}_1, \dots, \mathbf{h}_1, \dots, \mathbf{h}_L\}$.

However, unlike standard GP, in DGP the intermediate nodes are latent variables *i.e.* not observable, which makes the analytical computation of the marginal likelihood intractable. This is due to the integration of the conditional probability $p(\mathbf{h}_{l+1}|\mathbf{h}_l)$ containing the latent variable $\mathbf{h}_l$ non-linearly inside the inverse of the covariance matrix $K_{\mathbf{h}_l\mathbf{h}_l} + \sigma_l^2\mathbf{I}$.

To overcome this issue a variational tractable lower bound of the marginal likelihood is approximated [17]. This is accomplished in two steps. First, by introducing inducing variables in each layer. Inducing variables were first introduced in the context of sparse GP [25] [26]. It consists in augmenting with additional input-output pairs $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ and $\mathbf{u} = f(\mathcal{Z})$, the latent space, where $M << N$. This approach avoids the computation of the inverse of the covariance

matrix of the whole dataset $K_{\mathcal{X},\mathcal{X}} \in \mathcal{M}_{NN}$ and instead the inverse of the covariance matrix of the inducing inputs is computed $K_{\mathcal{Z},\mathcal{Z}} \in \mathcal{M}_{MM}$, hence, achieving reduction in the computational complexity in the training and prediction of a GP. In DGP, inducing variables $\mathcal{Z}_l = \{\mathbf{z}_{l1}, \ldots, \mathbf{z}_{lM_l}\}$ and $\mathbf{u}_l = \mathbf{f}_l(\mathcal{Z}_l)$ are introduced in each layer (Fig. 6). Then, by marginalizing the variables $\{\mathbf{u}_l\}_1^L$ the marginal likelihood can be rewritten as:

$$p(\mathbf{y}|\mathcal{X}) = \int_{\{\mathbf{h}_l,\mathbf{u}_l\}_1^L} p\left(\mathbf{y}, \{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L | \mathcal{X}, \{\mathcal{Z}_l\}_1^L\right) \mathrm{d}\{\mathbf{h}_l\}_1^L \mathrm{d}\{\mathbf{u}_l\}_1^L \tag{7}$$

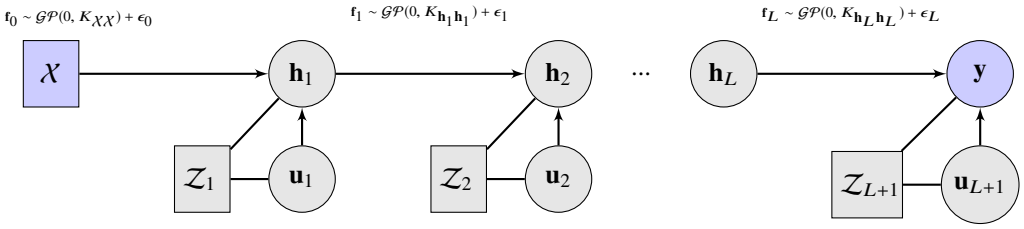


Fig. 6 Representation of the introduction of the inducing variables in DGPs

Next, by using the same variational approach used in [27] consisting of approximating the joint distribution of the true posterior of the latent variables $\mathbf{u}_l$ and $\mathbf{h}_l$ by multivariate Gaussian variational distributions $q(\mathbf{u}_l, \mathbf{h}_l)$ with the assumption of independency between layers [17]:

$$q\left(\{\mathbf{h}_l, \mathbf{u}_l\}_1^L\right) = \prod_{l=1}^{L} q(\mathbf{h}_l) q(\mathbf{u}_l)$$

By introducing this approximation of the posterior in the expression of $\log p(\mathbf{y}|\mathcal{X})$ and using Jensen's inequality, a variational lower bound on the marginal likelihood is obtained:

$$\begin{aligned} \log p(\mathbf{y}|\mathcal{X}) &= \log \int_{\{\mathbf{h}_l,\mathbf{u}_l\}_1^L} \frac{q(\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L)}{q(\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L)} p\left(\mathbf{y}, \{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L | \mathcal{X}, \{\mathcal{Z}_l\}_1^L\right) \mathrm{d}\{\mathbf{h}_l\}_1^L \mathrm{d}\{\mathbf{u}_l\}_1^L \\ &\geq \mathbb{E}_{q(\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L)} \left[\log \frac{p\left(\mathbf{y}, \{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L | \mathcal{X}, \{\mathcal{Z}_l\}_1^L\right)}{q(\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L)}\right] = \mathcal{L} \end{aligned} \tag{8}$$

After using some results from variational sparse GP [26] an analytical tractable bound is obtained for kernels that are feasibly convoluted with the Gaussian density such as the Automatic Relevance Determination (ARD) exponential kernel. The analytical optimal form of $q(\mathbf{u}_l)$ as a function of $q(\mathbf{h}_l)$ can be obtained *via* the derivative of the variational lower bound $\mathcal{L}$ w.r.t $q(\mathbf{u}_l)$. Hence, collapsing $q(\mathbf{u}_l)$ in the approximation by injecting its optimal form and obtaining a tighter lower bound depending on the following parameters:

- The kernel parameters: $\{\Theta_l\}_{l=1}^{l=L}$
- The inducing inputs $\{\mathcal{Z}_l\}_{l=1}^{l=L}$
- The variational distributions parameters $\{q(\mathbf{h}_l) \sim \mathcal{N}(\mathbf{m}_l, \mathbf{S}_l)\}_{l=1}^{l=L}$

Therefore, training a DGP model comes back to maximizing the evidence lower bound with respect to these parameters:

$$\begin{aligned} &\text{Maximize:} && \mathcal{L} \\ &\text{According to:} && \{\Theta_l\}_{l=1}^{l=L}, \{\mathcal{Z}_l\}_{l=1}^{l=L}, \{\mathbf{m}_l\}_{l=1}^{l=L}, \{\mathbf{S}_l\}_{l=1}^{l=L} \end{aligned}$$

Hence, the number of hyperparameters to optimize in the training of a DGP is more important than regular GP where only the kernel hyperparameters are considered. Alternative methods for training a DGP have been proposed. Dai *et al.* [28] instead of considering the parameters of the variational posteriors $q(\mathbf{h}_l)$ as individual parameters, considered them as a transformation of observed data $\mathcal{Y}$ by multi-layers perceptron. Bui *et al.* [29] proposed a deterministic approximation for DGPs based on an approximated Expectation Propagation energy function, and a probabilistic back-propagation algorithm for learning. The Doubly Stochastic approach proposed by Salimbeni *et al.* [30] drops

the assumption of independence between layers and the special form of kernels. Indeed, the posterior approximation maintains the exact model conditioned on $\mathbf{u}_l$:

$$q\left(\{\mathbf{h}_l, \mathbf{u}_l\}_1^L\right) = \prod_{l=1}^{L} p(\mathbf{h}_l|\mathbf{h}_{l-1}, \mathbf{u}_l) q(\mathbf{u}_l)$$

However, this costs the analytical tractability of the lower bound $\mathcal{L}$. The variational lower bound is then rewritten as follows (the mention of the dependence on $\mathcal{X}$ and $\mathcal{Z}$ is omitted for simplicity):

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q(\{\mathbf{h}_l, \mathbf{u}_l\}_1^L)}\left[\log \frac{p\left(\mathbf{y}, \{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L\right)}{q(\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L)}\right] \\
&= \mathbb{E}_{q(\{\mathbf{h}_l, \mathbf{u}_l\}_1^L)}\left[\log \frac{p\left(\mathbf{y}|\{\mathbf{h}_l\}_1^L, \{\mathbf{u}_l\}_1^L\right) \prod_{l=1}^{L} p(\mathbf{h}_l|\mathbf{h}_{l-1}, \mathbf{u}_l) p(\mathbf{u}_l)}{\prod_{l=1}^{L} p(\mathbf{h}_l|\mathbf{h}_{l-1}, \mathbf{u}_l) q(\mathbf{u}_l)}\right] \\
&= \mathbb{E}_{q(\{\mathbf{h}_l, \mathbf{u}_l\}_1^L)}\left[\log \frac{\prod_{i=1}^{N} p(y^{(i)}|\mathbf{f}_L^{(i)}) \prod_{l=1}^{L} p(\mathbf{u}_l)}{\prod_{l=1}^{L} q(\mathbf{u}_l)}\right] \\
\mathcal{L} &= \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{h}_L^{(i)})}\left[\log p(y^{(i)}|\mathbf{h}_L^{(i)})\right] - \sum_{l=1}^{L} KL\left[q(\mathbf{u}_l||p(\mathbf{u}_l)\right]
\end{aligned} \tag{9}$$

This formulation of the variational lower bound allows factorization over the data $\mathcal{X}, \mathcal{Y}$ which enable parallelization. The computation of this bound is done by approximating the expectation with Monte Carlo sampling, which is straightforward using the propagation of each data-point $\mathbf{x}^{(i)}$ through all the GPs:

$$q(\mathbf{h}_L^{(i)}) = \int \prod_{l=1}^{L-1} q\left(\mathbf{h}_l^{(i)}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, \mathbf{h}_{l-1}^{(i)}, \mathcal{Z}_{l-1}\right) \mathrm{d}\mathbf{h}_l^{(i)}$$

with $\mathbf{h}_0^{(i)} = \mathbf{x}^{(i)}$. The optimization of this formulation of the bound is done according to:

- The kernel parameters: $\{\Theta_l\}_{l=1}^{l=L}$
- The inducing inputs $\{\mathcal{Z}_l\}_{l=1}^{l=L}$
- The variational distributions of the inducing variables: $\{q(\mathbf{u}_l) \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\}_{l=1}^{l=L}$

B. DGPs and Bayesian Optimization

The deep architecture of a DGP increases the model capability compared to a simple GP allowing the capturing of non-stationary phenomena (Fig. 7, 8). Hence, its coupling with BO to handle the optimization of non-stationary functions is interesting. In fact, for single objective optimization problems, experimentations in [1] show that BO coupled with DGPs outperform standard BO (coupled with GPs) and BO with non-linear mapping. In [2] a more thorough investigation on the coupling of BO with DGPs is given. In this current work the attention is paid to the multi-objective case. Focusing on the training approach, the infill criteria and the configuration of the architecture.

- **Training approach:** Multiple approaches have been developed for training DGPs as discussed previously. In the first attempt to use DGPs for BO in [1] the auto-encoded variational approach was used for training. However, in [2] the doubly stochastic variational approach is used to keep the dependency between layers making this approach more robust. The experimental results of BO with DGP using this training approach confirms this choice by giving more robust results especially when the architecture of the DGP gets deeper [2]. Since, in BO the objective is to reduce the time in the optimization, one can not train in each iteration the model multiple times until obtaining the best model. So, the most robust approach of training is preferred.
- **Infill criteria:** In single objective BO with GPs, infill criteria such as the Expected Improvement, the Probability of Improvement or the Expected Violation are computed using closed analytic formulae. These formulae are obtained based on the Gaussian distribution of the Gaussian Process prediction. However, in DGPs the overall process prediction is no longer Gaussian. Thus, in order to use a valid approximation of the infill criteria, it is necessary to approximate the distribution of the prediction by a Gaussian distribution, and if not, to use a sampling approach on the value of the prediction [2]. In the multi-objective case the closed form analytic equation of the

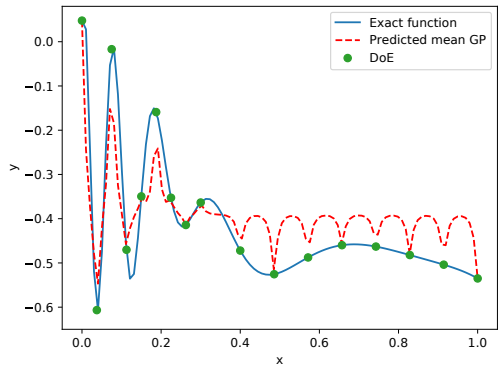


Fig. 7 Approximation of the modified-Xiong function by a regular GP. The model can not capture the stability of the region $[0.4, 1]$ **and continues to oscillate**

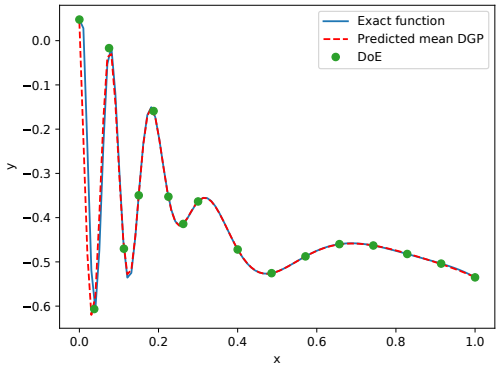


Fig. 8 Approximation of the modified-Xiong function by a DGP. The DGP model appropriately capture the two regions with different smoothness

EHVI in Eq. 4 is also obtained with the assumption that the prediction of the objective functions follows a normal distribution. Hence, the same approximations in the prediction used for the EI are necessary for the EHVI. The same prediction scheme used in [2] is followed here (Fig.9).

- **Configuration of the architecture:** Discussing the architecture of the DGPs concerns the number of layers, the number of hidden units at each layer and the number of induced inputs at each layer. The DGPs tend to perform better (in terms of prediction and robustness) when increasing these architecture variables as observed in [2]. However, the configuration of the architecture directly influences the computational complexity of the evaluation of the evidence lower bound $\mathcal{L}$ given by $O(N(M_1^2 D_1 + \ldots + M_l^2 D_l + \ldots + M_L^2 D_L))$, where N is the size of the data-set, L is the number of layers, M_l is the number of induced inputs at the layer l and D_l is the number of hidden units at layer l. This is more expensive in the multi-objective case when multiple objectives have to be approximated. Therefore, a trade-off between the performance and the computational cost has to be found. Moreover, the particularity of using DGPs in a BO framework is that the number of datapoints changes at each iteration. Thus, the configuration of the architecture has to be adapted to the current iteration. In fact, in the early iterations when the datasize is small a simple architecture (a standard GP, a 1-layer DGP) is sufficient. Then, along the evolution of the size of the dataset a more complex architecture can be developed. If the stationary behavior is known *a priori* for some objective functions or constraints, one can use only GPs for some functions while using DGPs for the unknown or non-stationary functions.

V. Analytical Experimentation

In this section, experimentations on an analytical test problem are performed to compare standard MO-BO using GPs, NSGA II, and MO-BO using DGPs.

A. Problem

The analytical test case is a two-objective problem with a non-stationary constraint. The problem (P_1) has been inspired by the TNK test problem [31] with a modification of the constraint making it non stationary. In fact, there are two regions, one where the function varies with a high frequency and another one where the function has small variations (Fig. 10).

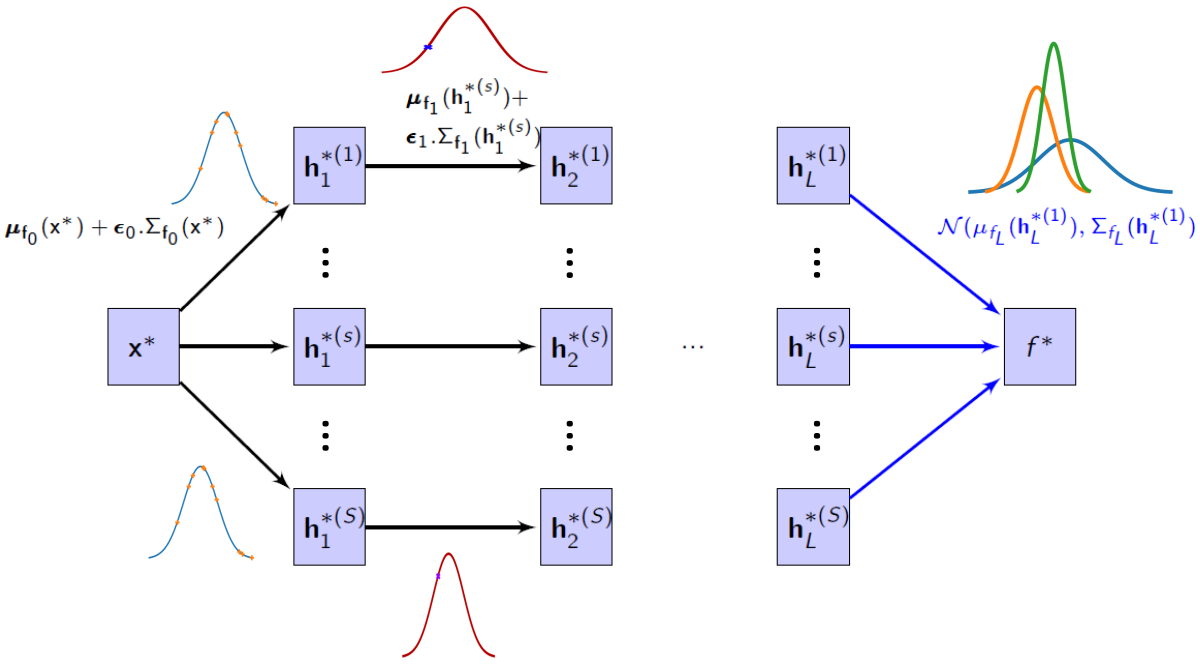


Fig. 9 The approximation of the prediction of a DGP model by a mixture of Gaussian distribution. S samples are drawn from the first layer, then, each sample is propagated through the whole network, with a realization at each hidden layer, until reaching the final layer where the mean and the variance of the final GP are considered for each sample. Thus, the prediction is approximated by a Gaussian mixture of the S samples.

$$
P_1 \left| \begin{array}{lll} \text{Min} & f_1(\mathbf{x}) & = -x_1 \\ \text{Min} & f_2(\mathbf{x}) & = -x_2 \\ \text{s.t} & g_1(\mathbf{x}) & = 0.5x_1^2 + 0.5x_2^2 - 0.2\cos(20\arctan(0.3\frac{x_1}{x_2})) \leq 0 \\ \text{with} & & \mathbf{x} = [x_1, x_2] \\ \text{and} & & 0 < x_1 < 1 \\ \text{and} & & 0 < x_2 < 1 \end{array} \right. \tag{10}
$$

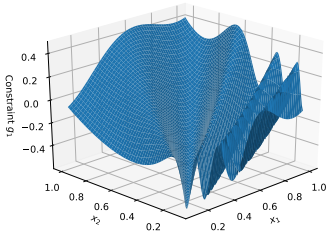


Fig. 10 Constraint function

The Pareto front given by this problem has three separated regions (Fig. 11 and 12). The reference value of the hypervolume dominated in the rectangle $[[-1,-1],[0,0]]$ is 0.752.

B. Parameter settings

For NSGA-II, an initial population of 5 individuals is generated and the algorithm is run until 45 evaluations are reached. For standard MO-BO and MO-BO with DGPs, 25 points are generated using a Latin Hypercube Sampling and 30 points are added using the EHVI with the probability of feasibility optimized with a Differential Evolution algorithm [32]. To evaluate the robustness of each algorithm the experimentation is repeated for 10 different initial DoE.

- In NSGA-II, a simulated binary crossover is used, with a distribution index=15 and a probability of 0.9, and a polynomial mutation with a distribution index of 20 and a probability of 1/6. The constraint dominance is used to handle the constraints.

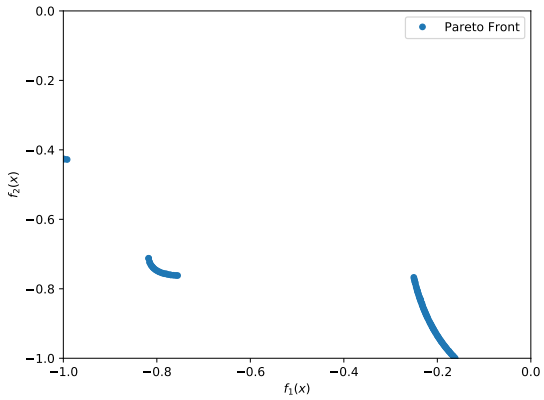


Fig. 11 Exact Pareto Front

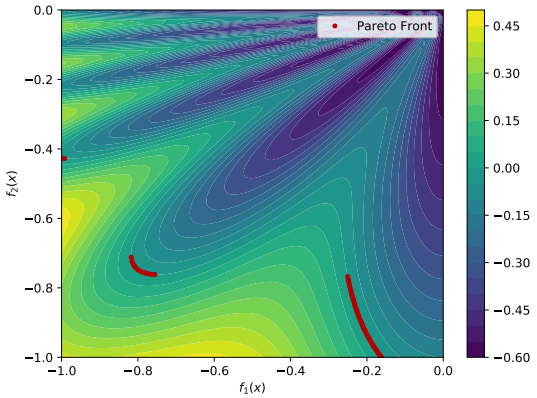


Fig. 12 Exact Pareto Front with constraint contour-plot

- In standard MO-BO, an Automatic Relevance Determination (ARD) exponential kernel [8] is used: $k(\mathbf{x}, \mathbf{x'}) = \exp\{-\sum_{i=1}^{D} \theta_i (x_i - x'_i)^2\}$.
- In MO-BO with DGP, only the constraint is approximated by a DGP since the objective functions are stationary. An ARD Gaussian kernel is used in each layer. The training of the DGP is performed using the Doubly Stochastic training approach [30]. Configurations with 1, 2 and 3 layers are tested with a number of induced inputs equal to the dataset size. The number of units in the hidden layers is fixed to 6. The prediction is approximated with 500 samples.

C. Experimental results

Table 1 displays the median of the hypervolume value on the 10 repetitions and its corresponding first and third quartiles at the end of each algorithm (45 evaluations). Fig. 14 gives the Pareto front of each algorithm for each repetition. The plots of convergence of the BO algorithms are displayed in Fig. 13.

Table 1 Performance of the algorithms

Algorithm	**Hypervolume median**	**1^{st} quartile Hypervolume**	**3^{rd} quartile Hypervolume**
NSGA-II	0.485	0.186	0.664
MO-BO GP	0.682	0.664	0.700
MO-BO DGP 1HL	0.737	0.716	0.743
MO-BO DGP 2HL	0.738	0.715	0.744
MO-BO DGP 3HL	0.739	0.726	0.741

As expected NSGA-II is the algorithm which performs less efficiently. In fact, NSGA-II needs more evaluations to give appropriate results and with only 45 evaluations the algorithm is far from convergence, which explains the high scattering of the Pareto fronts according to the repetitions. It happens that BO with GP gives good results in some repetitions, however it has an important variance among the repetitions. This behavior can be explained by the fact that the initial DoE for the worst repetitions is concentrated in the region of high frequency and so the Gaussian process can not capture the region of low frequency and *vice versa*. BO with DGPs performs clearly better than regular GP regardless of the number of layers considered. It is also robust to the initial DoE as shown in the plots of the Pareto fronts where each repetition reach with a remarkable accuracy the exact Pareto front. The convergence plot of the

different BO shows a separation between BO with GP and with DGPs, and this can be noticed since the early iterations. The trade-off between computational complexity in the training of a DGP and the power of representation is important to be considered. In fact, in this problem there is no clear difference between the three considered configurations of DGPs. Hence, the capacity of the DGP with only one layer is sufficient to capture the non-stationarity of this problem, and there is no need to go deeper.

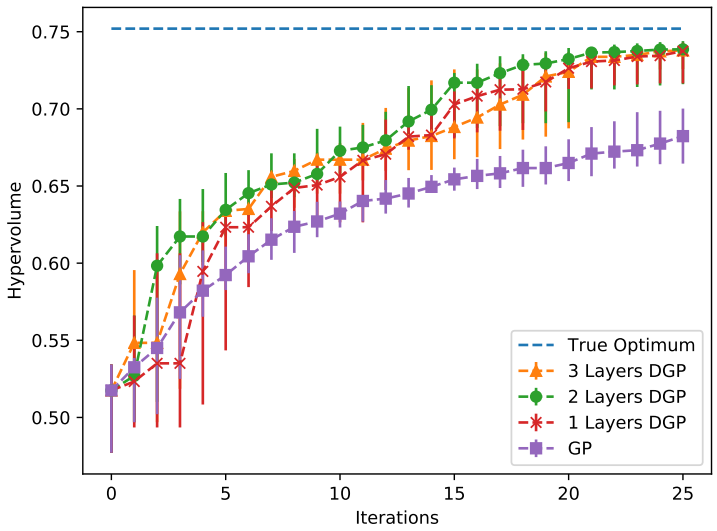
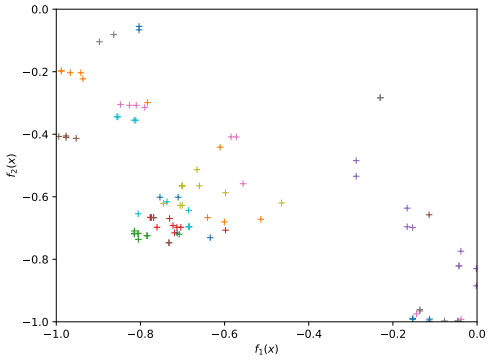
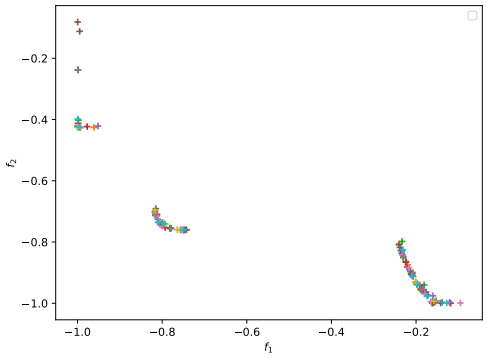


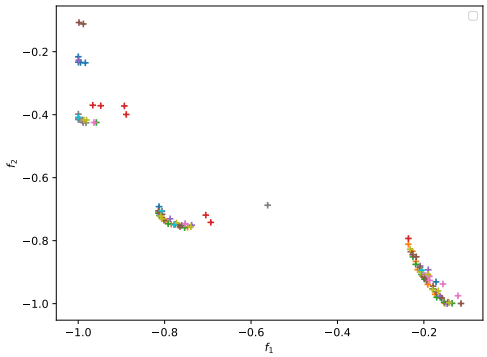
Fig. 13 Convergence plot of BO with different architectures of DGPs and a regular GP. The markers indicate the median of the hypervolume obtained while the errorbars indicate the first and the third quartiles.



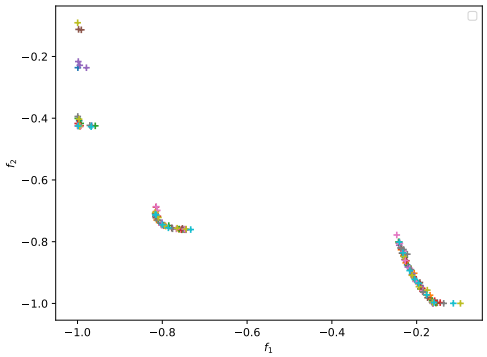
NSGA-II Pareto Fronts



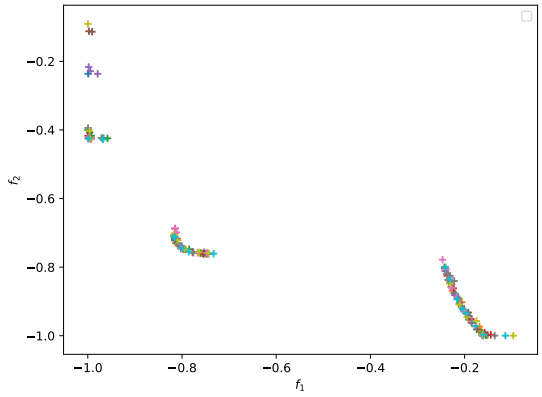
DGP 1HL Pareto Fronts



Standard EGO Pareto Fronts



DGP 2HL Pareto Fronts



DGP 3HL Pareto Fronts

Fig. 14 Pareto Fronts of the different repetitions for each algorithm. Each repetition corresponds to a certain color.

VI. Aerospace vehicle design optimization

To confirm the interest of the MO-BO and DGP approach, an aerospace vehicle design optimization problem is considered consisting of the optimization of a set of objectives for a solid-propellant booster engine. It is a representative physical problem for solid booster design with simulation models fast enough to provide the exact Pareto front to compare and illustrate the efficiency of the proposed algorithms.

A. Description of the problem

The optimization of a set of objectives for a solid propellant booster is considered (Fig 15). The objectives are:

- Minimization of the Gross Lift-off Weight (GLOW)
- Maximization of the change in velocity (ΔV)

In addition, four design variables are considered:

- Propellant mass: 5 t < $\mathbf{M}_{prop}$ < 15 t
- Combustion chamber pressure: 5 bar < $\mathbf{P}_c$ < 100 bar
- Throat nozzle diameter: 0.2 m < $\mathbf{D}_c$ < 1 m
- Nozzle exit diameter: 0.5 m < $\mathbf{D}_s$ < 1.2 m

Different constraints are also considered including a structural one limiting the combustion pressure according to the motor case, 6 geometrical constraints on the internal vehicle layout for the propellant and the nozzle, and a jet breakaway constraint concerning the throat nozzle diameter and the nozzle exit diameter.

$$
\begin{aligned}
&\text{Minimize:} && [GLOW(\mathbf{X}), -\Delta V(\mathbf{X})] \\
&\text{According to:} && \mathbf{X} = [\mathbf{M}_{prop}, \mathbf{P}_c, \mathbf{D}_c, \mathbf{D}_s] \\
&\text{subject to:} && \begin{cases} \text{1 structural constraint} \\ \text{6 geometrical constraints} \\ \text{1 jet breakaway constraints} \end{cases}
\end{aligned}
$$

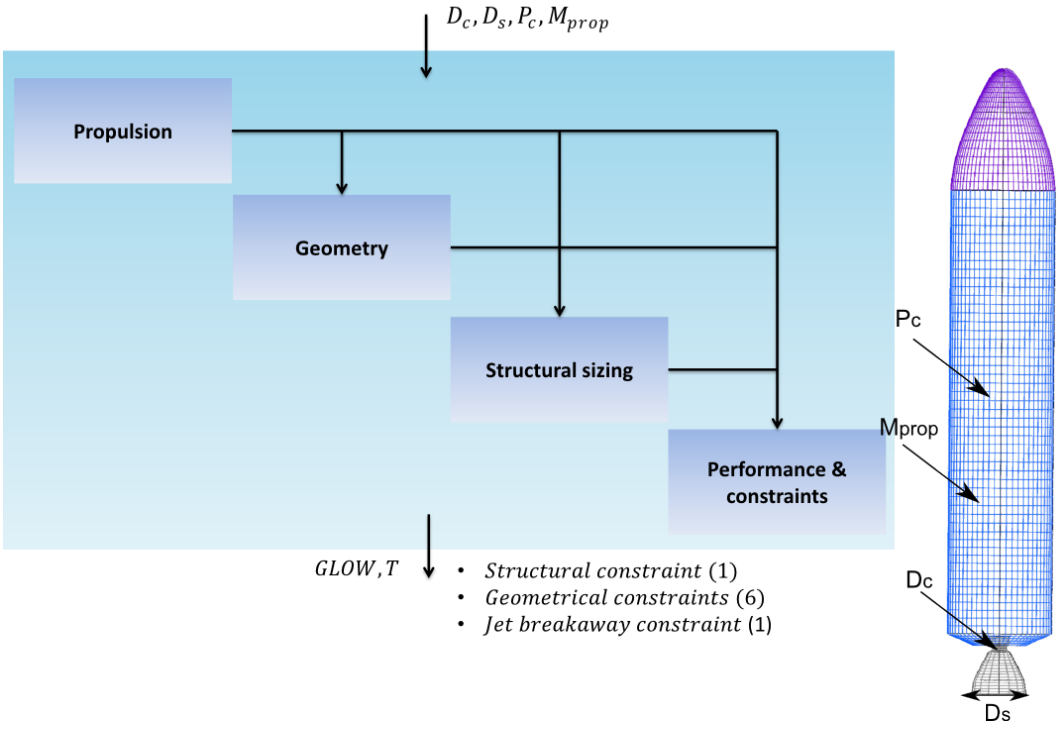
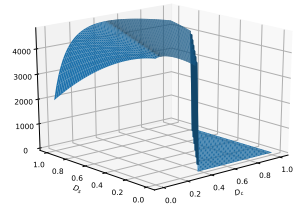
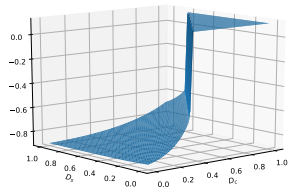


Fig. 15 Two-stage booster vehicle design multi-objective optimization

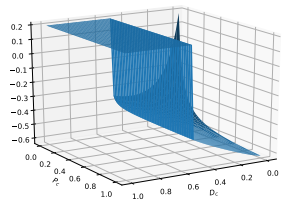
This problem is expected to have non-stationarity behaviors due to some constraints. In fact, the constraints may have a different behavior in the feasible and unfeasible regions. Moreover, the objective functions may also be difficult



A sectional view of the change in velocity according to the diameters of the nozzle.



A sectional view of a constraint according to the diameters of the nozzle.



A sectional view of a constraint according to the throat nozzle diameter and the combustion chamber pressure.

Fig. 16 Sectional view of the non-stationary behaviors of some functions involved in the booster problem

to be approximated *via* simple GPs, for example the change in velocity function may have a tray region when it is equal to zero, due to an insufficient propellant mass (Fig. 16). Hence, to obtain an approximated Pareto front for this problem a MO-BO approach using DGPs (1 and 2 layer configurations) is chosen. The number of units in the hidden layers is fixed to 6 and the number of induced inputs is equal to the dataset size in each iteration. The results obtained are compared to BO with standard GP and NSGA-II.

B. Experimental results

The initial DoE are set using a Latin Hypercube Sampling of 40 points and 60 points are added with BO. Five repetitions are performed to assess the robustness of the results.

The plots of convergence of the BO algorithms are displayed in Fig. 17. The first observation is that after adding 60 points the different BO algorithms either with GP or DGPs converge toward the same hypervolume value, with a slight advantage for the DGP BO. However, it is interesting to point out that the speed of convergence of BO with DGP is clearly better than BO with simple GPs. Actually, after adding only 20 points the BO with 2 hidden layers has almost converged with a better robustness to the initial DoE (see Table 2). As expected NSGA-II with the same number of evaluations as BO is not able to converge to the same hypervolume and is subject to an important variation.

Fig. 18 displays the evolution of the approximated Pareto front given by BO with DGP 2 hidden layers, and the final Pareto front after 60 added point is compared to the Pareto front given by NSGA-II with 1000 population and 100000 evaluations. The final Pareto front obtained is a continuous arc with a change in velocity ΔV varying between $5000m/s$ and $3600m/s$, and a Gross Lift Off Weight varying between $6t$ and $14t$. It is interesting to point out than even with a huge number of evaluations the approximated Pareto front given by NSGA-II does not dominate the approximated Pareto front given by BO with DGP 2HL after only 60 added points. Indeed, the region of the objective space with $4900m/s \leq \Delta V \leq 5000m/s$ and $12t \leq GLOW \leq 14t$ is better approximated by BO with DGP.

Table 2 Performance of the algorithms after 20 added points (60 evaluations for NSGA-II) and after 60 added points (100 evaluations for NSGA-II). HL stands for hidden layer.

	After 20 added points (60 evaluations NSGA-II)			After 60 added points (100 evaluations NSGA-II)		
Algorithm	**Average Hypervolume**	**Max hypervolume**	**min Hypervolume**	**Average Hypervolume**	**Max hypervolume**	**min Hypervolume**
NSGA-II	0.576	0.6761	0.4923	0.611	0.73082	0.5184
MO-BO GP	0.7917	0.811	0.772	0.82643	0.8302	0.821
MO-BO DGP 1HL	0.8068	0.8234	0.775	0.8328	0.8342	0.8317
MO-BO DGP 2HL	0.8153	0.8236	0.8049	0.8321	0.8347	0.8278

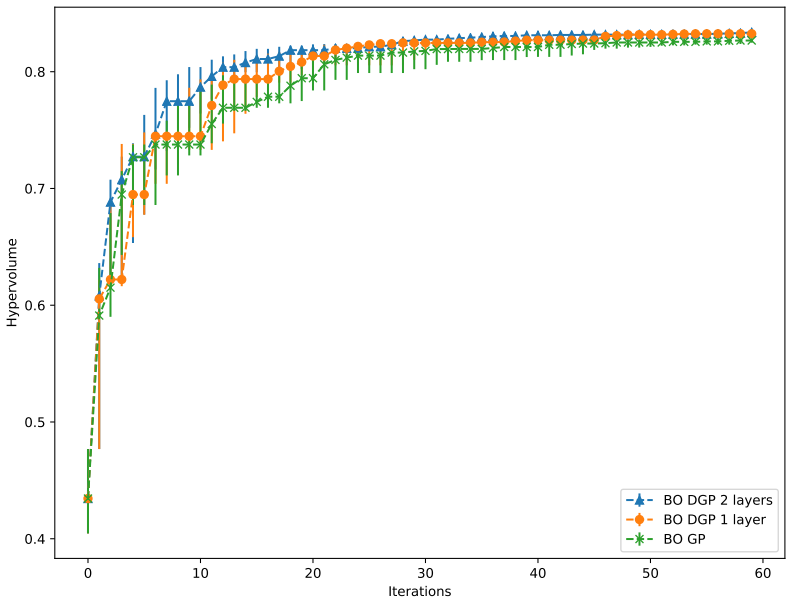
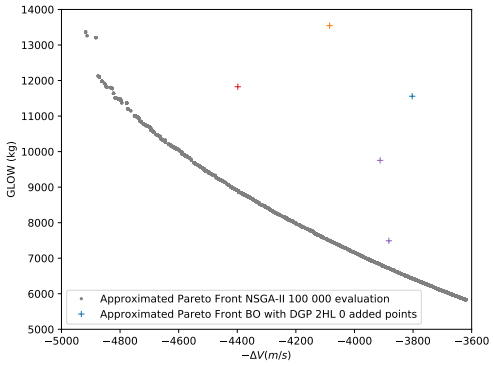
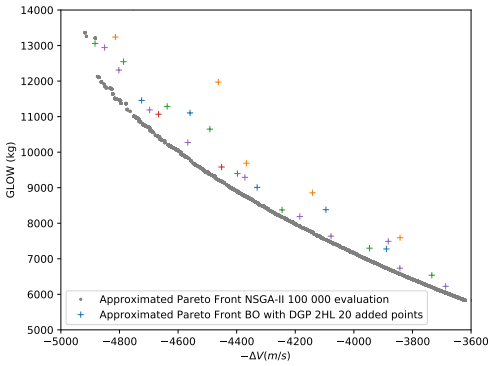


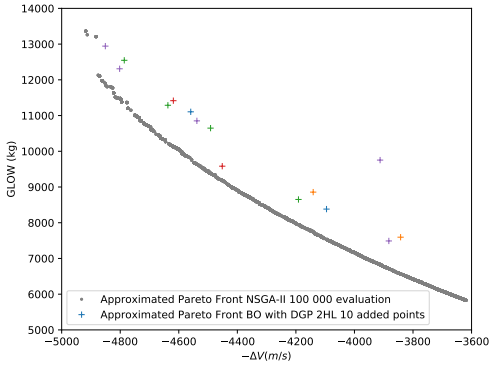
Fig. 17 Convergence plot of BO with different architectures of DGPs and a regular GP. The markers indicate the median of the hypervolume obtained while the errorbars indicate the minimum and maximum.



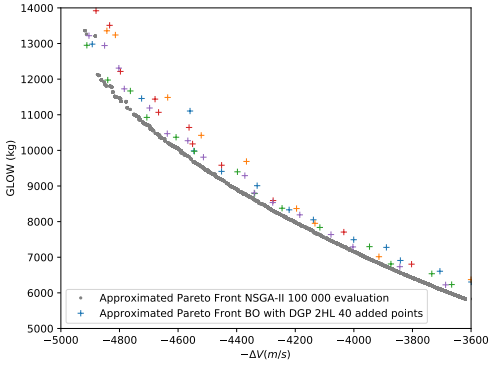
Initial DoE 40 initial points (0 Added points)



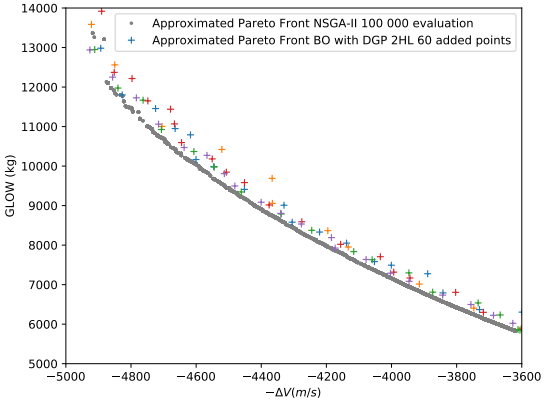
20 Added points



10 Added points



40 Added points



60 Added points

Fig. 18 Evolution of the approximated Pareto Fronts of the different repetitions of BO with DGP 2HL. Each repetition corresponds to a certain color.

VII. Conclusions

In this paper, the coupling of MO-BO with DGPs has been discussed and applied to an analytical test case and an aerospace vehicle design problem demonstrating the interest of the proposed approach. Indeed, in each of the performed experiments the MO-BO with a DGP configuration performs better, converges faster and is more robust to the initial DoE than MO-BO with a standard GP. The main drawback of the DGP approach may be the setting of its configuration. Indeed, one has to balance between how deep can the network gets to obtain more precision and the computation time in the training of the model. In this work, a DGP with only one hidden layer was sufficient to obtain good results even if a slight improvement is observed when increasing the number of layers.

In this context, future works may concern the development of an adaptive framework for the configuration of the DGP according to the problem at hand. Also the time of training the DGP can be problematic with complex models, an interesting direction of research is to investigate ways to accelerate the training process. Finally, here in the multi-objective case the objectives were considered independent, one may gain some information by creating a dependence between the objectives using the concept of multi-output GPs and co-regionalization.

Acknowledgments

This work was co-funded by ONERA-The French Aerospace Lab and Université de Lille Lille, in the context of a joint PhD thesis. In addition, Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

References

[1] Hebbal, A., Brevault, L., Balesdent, M., Taibi, E.-G., and Melab, N., "Efficient Global Optimization using Deep Gaussian Processes," *2018 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2018, pp. 1–8.

[2] Hebbal, A., Brevault, L., Balesdent, M., Talbi, E.-G., and Melab, N., "Bayesian Optimization using Deep Gaussian Processes for Non-Stationary Problems," *arXiv preprint arXiv:1809.04632*, 2018.

[3] Arias-Montano, A., Coello, C. A. C., and Mezura-Montes, E., "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Transactions on Evolutionary Computation*, Vol. 16, No. 5, 2012, pp. 662–694.

[4] Deb, K., *Multi-objective optimization using evolutionary algorithms*, Vol. 16, John Wiley & Sons, 2001.

[5] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *International Conference on Parallel Problem Solving From Nature*, Springer, 2000, pp. 849–858.

[6] Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello, C. C., Luna, F., and Alba, E., "Smpso: A new pso-based metaheuristic for multi-objective optimization," *Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on*, IEEE, 2009, pp. 66–73.

[7] Wang, G., and Shan, S., "Review of metamodeling techniques in support of engineering design optimization," *Journal of Mechanical design*, Vol. 129, No. 4, 2007, pp. 370–380.

[8] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[9] Rasmussen, C., and Williams, C. K., *Gaussian processes for machine learning*, Vol. 1, MIT press Cambridge, 2006.

[10] Beume, N., Naujoks, B., and Emmerich, M., "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, Vol. 181, No. 3, 2007, pp. 1653–1669.

[11] Wagner, T., Emmerich, M., Deutz, A., and Ponweiser, W., "On expected-improvement criteria for model-based multi-objective optimization," *International Conference on Parallel Problem Solving from Nature*, Springer, 2010, pp. 718–727.

[12] Higdon, D., Swall, J., and Kern, J., "Non-stationary spatial modeling," *Bayesian statistics*, Vol. 6, No. 1, 1999, pp. 761–768.

[13] Paciorek, C. J., and Schervish, M. J., "Spatial modelling using a new class of nonstationary covariance functions," *Environmetrics*, Vol. 17, No. 5, 2006, pp. 483–506.

[14] Haas, T. C., "Kriging and automated variogram modeling within a moving window," *Atmospheric Environment. Part A. General Topics*, Vol. 24, No. 7, 1990, pp. 1759–1769.

[15] Rasmussen, C. E., and Ghahramani, Z., "Infinite mixtures of Gaussian process experts," *Advances in neural information processing systems*, 2002, pp. 881–888.

[16] Xiong, Y., Chen, W., Apley, D., and Ding, X., "A non-stationary covariance-based Kriging method for metamodelling in engineering design," *International Journal for Numerical Methods in Engineering*, Vol. 71, No. 6, 2007, pp. 733–756.

[17] Damianou, A., and Lawrence, N., "Deep gaussian processes," *Artificial Intelligence and Statistics*, 2013, pp. 207–215.

[18] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering optimization*, Vol. 34, No. 3, 2002, pp. 263–278.

[19] Emmerich, M. T., Giannakoglou, K. C., and Naujoks, B., "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 4, 2006, pp. 421–439.

[20] Knowles, J., "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 1, 2006, pp. 50–66.

[21] Zhang, Q., Liu, W., Tsang, E., and Virginas, B., "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, Vol. 14, No. 3, 2010, pp. 456–474.

[22] Svenson, J. D., and Santner, T. J., "Multiobjective optimization of expensive black-box functions via expected maximin improvement," *The Ohio State University, Columbus, Ohio*, Vol. 32, 2010.

[23] Emmerich, M., and Klinkenberg, J.-w., "The computation of the expected improvement in dominated hypervolume of Pareto front approximations," *Rapport technique, Leiden University*, Vol. 34, 2008.

[24] Bader, J., and Zitzler, E., "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary computation*, Vol. 19, No. 1, 2011, pp. 45–76.

[25] Snelson, E., and Ghahramani, Z., "Sparse Gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[26] Titsias, M., "Variational learning of inducing variables in sparse Gaussian processes," *Artificial Intelligence and Statistics*, 2009, pp. 567–574.

[27] Titsias, M., and Lawrence, N. D., "Bayesian Gaussian process latent variable model," *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 844–851.

[28] Dai, Z., Damianou, A., González, J., and Lawrence, N., "Variational auto-encoded deep Gaussian processes," *arXiv preprint arXiv:1511.06455*, 2015.

[29] Bui, T., Hernández-Lobato, D., Hernandez-Lobato, J., Li, Y., and Turner, R., "Deep gaussian processes for regression using approximate expectation propagation," *International Conference on Machine Learning*, 2016, pp. 1472–1481.

[30] Salimbeni, H., and Deisenroth, M., "Doubly Stochastic Variational Inference for Deep Gaussian Processes," *arXiv preprint arXiv:1705.08933*, 2017.

[31] Deb, K., Pratap, A., and Meyarivan, T., "Constrained test problems for multi-objective evolutionary optimization," *International conference on evolutionary multi-criterion optimization*, Springer, 2001, pp. 284–298.

[32] Qin, A. K., Huang, V. L., and Suganthan, P. N., "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, Vol. 13, No. 2, 2009, pp. 398–417.